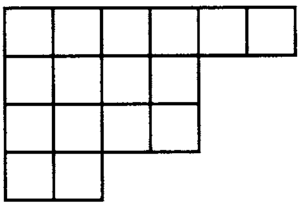
Notation

A ***Young diagram*** is a collection of boxes, or cells, arranged in left-justified rows, with a (weakly) decreasing number of boxes in each row. Listing the number of boxes in each row gives a partition of the integer n that is the total number of boxes. Conversely, every partition of n corresponds to a Young diagram. For example, the partition of 16 into $6+4+4+2$ corresponds to the Young diagram



We usually denote a partition by a lowercase Greek letter, such as λ. It is given by a sequence of weakly decreasing positive integers, sometimes written $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$; it is often convenient to allow one or more zeros to occur at the end, and to identify sequences that differ only by such zeros. One sometimes writes $\lambda = ({d_1}^{a_1} \ldots {d_s}^{a_s})$ to denote the partition that has a_i copies of the integer d_i, $1 \leq i \leq s$. The notation $\lambda \vdash n$ is used to say that λ is a partition of n, and $|\lambda|$ is used for the number partitioned by λ. We usually identify λ with the corresponding diagram, so we speak of the second row, or the third column, of λ.

The purpose of writing a Young diagram instead of just the partition, of course, is to put something in the boxes. Any way of putting a positive integer in each box of a Young diagram will be called a ***numbering*** or ***filling*** of the diagram; usually we use the word *numbering* when the entries are distinct, and *filling* when there is no such requirement. A ***Young tableau***, or simply

tableau, is a filling that is

(1) weakly increasing across each row

(2) strictly increasing down each column

We say that the tableau is a tableau ***on*** the diagram λ, or that λ is the ***shape*** of the tableau. A ***standard tableau*** is a tableau in which the entries are the numbers from 1 to n, each occurring once. Examples, for the partition (6,4,4,2) of 16, are

1	2	2	3	3	5
2	3	5	5		
4	4	6	6		
5	6				

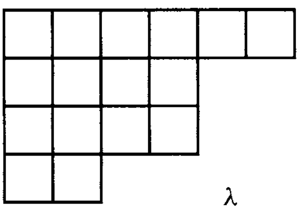
Tableau

1	3	7	12	13	15
2	5	10	14		
4	8	11	16		
6	9				

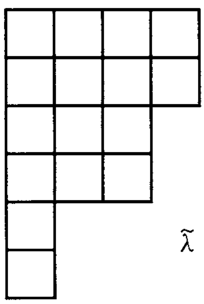
Standard tableau

Entries of tableaux can just as well be taken from any ***alphabet*** (totally ordered set), but we usually take positive integers.

Describing such combinatorial data in the plane also suggests simple but useful geometric constructions. For example, flipping a diagram over its main diagonal (from upper left to lower right) gives the ***conjugate*** diagram; the conjugate of λ will be denoted here by $\tilde{\lambda}$. As a partition, it describes the lengths of the columns in the diagram. The conjugate of the above partition is (4,4,3,3,1,1):



λ



$\tilde{\lambda}$

Any numbering T of a diagram determines a numbering of the conjugate, called the ***transpose***, and denoted T^{τ}. The transpose of a standard tableau is a standard tableau, but the transpose of a tableau need not be a tableau.

It may be time already to mention the morass of conflicting notation one will find in the literature. Young diagrams are also known as *Ferrers diagrams* or *frames*; sometimes dots are used instead of boxes, and sometimes, particularly in France, they are written upside down, in order not to offend Descartes. What we call tableaux are known variously as *semistandard* tableaux, or *column-strict* tableaux, or *generalized Young tableaux* (in which case our standard tableaux are just called *tableaux*). Combinatorialists also know them as *column-strict reversed plane partitions*; the "reversed" is in opposition to the case of decreasing rows and columns, which was studied first (and allows zero entries); cf. Stanley (1971).

Associated to each partition λ and integer m such that λ has at most m parts (rows), there is an important symmetric polynomial $s_\lambda(x_1, \ldots, x_m)$ called a ***Schur polynomial***. These polynomials can be defined quickly using tableaux. To any numbering T of a Young diagram we have a monomial, denoted x^T, which is the product of the variables x_i corresponding to the i's that occur in T. For the tableau in the first diagram, this monomial is $x_1 x_2{}^3 x_3{}^3 x_4{}^2 x_5{}^4 x_6{}^3$. Formally,

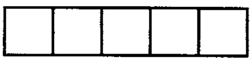
$$x^T = \prod_{i=1}^{m} (x_i)^{\text{number of times } i \text{ occurs in } T}.$$

The Schur polynomial $s_\lambda(x_1, \ldots, x_m)$ is the sum

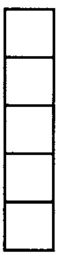
$$s_\lambda(x_1, \ldots, x_m) = \sum x^T$$

of all monomials coming from tableaux T of shape λ using the numbers from 1 to m. Although it is not obvious from this definition, these polynomials are symmetric in the variables $x_1, \ldots, x_m$, and they form an additive basis for the ring of symmetric polynomials. We will prove these facts later.

The Young diagram of $\lambda = (n)$ has n boxes in a row

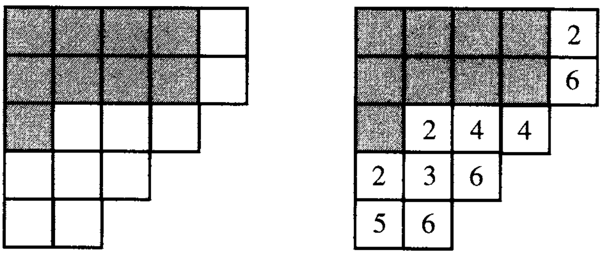


The Schur polynomial for this partition is the n^{th} ***complete symmetric polynomial***, which is the sum of all distinct monomials of degree n in the variables $x_1, \ldots, x_m$; this is usually denoted $h_n(x_1, \ldots, x_m)$. For the other extreme $n = 1 + \ldots + 1$, i.e., $\lambda = (1^n)$, the Young diagram is



The corresponding Schur polynomial is the n^{th} ***elementary symmetric polynomial***, which is the sum of all monomials $x_{i_1} \cdot \ldots \cdot x_{i_n}$ for all strictly increasing sequences $1 \le i_1 < \ldots < i_n \le m$, and is denoted $e_n(x_1, \ldots, x_m)$.

A ***skew diagram*** or ***skew shape*** is the diagram obtained by removing a smaller Young diagram from a larger one that contains it.[1] If two diagrams correspond to partitions $\lambda = (\lambda_1, \lambda_2, \ldots)$ and $\mu = (\mu_1, \mu_2, \ldots)$, we write $\mu \subset \lambda$ if the Young diagram of μ is contained in that of λ; equivalently, $\mu_i \le \lambda_i$ for all i. The resulting skew shape is denoted λ/μ. A ***skew tableau*** is a filling of the boxes of a skew diagram with positive integers, weakly increasing in rows and strictly increasing in columns. The diagram is called its ***shape***. For example, if $\lambda = (5,5,4,3,2)$ and $\mu = (4,4,1)$, the following shows the skew diagram λ/μ and a skew tableau on λ/μ:



The set $\{1, \ldots, m\}$ of the first m positive integers is denoted $[m]$.

[1] Algebraically, a collection of boxes is a skew shape if they satisfy the condition that when boxes in the (i, j) and (i', j') position are included, and $i \le i'$ and $j \le j'$, then all boxes in the (i'', j'') positions are included for $i \le i'' \le i'$ and $j \le j'' \le j'$.

Part I

Calculus of tableaux

There are two fundamental operations on tableaux from which most of their combinatorial properties can be deduced: the Schensted "bumping" algorithm, and the Schützenberger "sliding" algorithm. When repeated, the first leads to the Robinson–Schensted–Knuth correspondence, and the second to the "jeu de taquin." They are in fact closely related, and either can be used to define a product on the set of tableaux, making them into an associative monoid. This product is the basis of our approach to the Littlewood–Richardson rule.

In Chapter 1 we describe these notions and state some of the main facts about them. The proofs involve relations among words which are associated to tableaux, and are given in the following two chapters. Chapters 4 and 5 have the applications to the Robinson–Schensted–Knuth correspondence and the Littlewood–Richardson rule. See Appendix A for some of the many possible variations on these themes.

1

Bumping and sliding

1.1 Row-insertion

The first algorithm, called ***row-insertion*** or ***row bumping,*** takes a tableau T, and a positive integer x, and constructs a new tableau, denoted $T \leftarrow x$. This tableau will have one more box than T, and its entries will be those of T, together with one more entry labelled x, but there is some moving around. The recipe is as follows: if x is at least as large as all the entries in the first row of T, simply add x in a new box to the end of the first row. If not, find the left-most entry in the first row that is strictly larger than x. Put x in the box of this entry, and remove ("bump") the entry. Take this entry that was bumped from the first row, and repeat the process on the second row. Keep going until the bumped entry can be put at the end of the row it is bumped into, or until it is bumped out the bottom, in which case it forms a new row with one entry.

For example, to row-insert 2 in the tableau

1	2	2	3
2	3	5	5
4	4	6	
5	6		

the 2 bumps the 3 from the first row, which then bumps the first 5 from the second row, which bumps the 6 from the third row, which can be put at the end of the fourth row:

1	2	2	**3**	← 2
2	3	5	5	
4	4	6		
5	6			

1	2	2	2	
2	3	**5**	5	← 3
4	4	6		
5	6			

1	2	2	2	
2	3	3	5	
4	4	**6**		← 5
5	6			

1	2	2	2
2	3	3	5
4	4	5	
5	6	6	

It is clear from the construction that the result of this process is always a tableau. Indeed, each row is successively constructed to be weakly increasing, and, when an entry y bumps an entry z from a box in a given row, the entry below it, if there is one, is strictly larger than z (by the definition of a tableau), so z either stays in the same column or moves to the left, and the entry lying above its new position is no larger than y, so is strictly smaller than z.

There is an important sense in which this operation is invertible. If we are given the resulting tableau, *together with the location of the box that has been added to the diagram,* we can recover the original tableau T and the element x. The algorithm is simply run backwards. If y is the entry in the added box, it looks for its position in the row above the location of the box, finding the entry farthest to the right which is strictly less than y. It bumps this entry up to the next row, and the process continues until an entry is bumped out of the top row. This reverse bumping can be carried out for any tableau and any box in it that is an outside corner, i.e., a box in the Young diagram such that the boxes directly below and to the right are not in the diagram. For example, starting with the tableau and the shaded box

1	2	2	2
2	3	3	5
4	4	5	
5	6	**6**	

the 6 bumps the 5 in the third row, which bumps the right 3 in the second row, which bumps the right 2 from the first row – exactly reversing steps in the preceding example.

There is a simple lemma about the bumping algorithm which tells about the results of two successive bumpings, allowing one to relate the size of the elements inserted with the positions of the new boxes. A row-insertion $T \leftarrow x$ determines a collection R of boxes, which are those where an element is bumped from a row, together with the box where the last bumped element lands. Let us call this the ***bumping route*** of the row-insertion, and call the box added to the diagram for the last element the ***new box*** of the row-insertion. In the example, the bumping route consists of the shaded boxes, with the new box containing the 6:

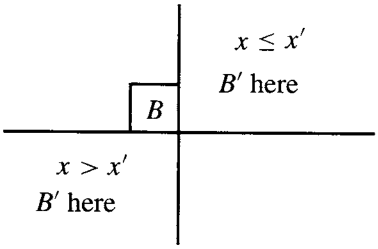
1	2	2	2
2	3	3	5
4	4	5	
5	6	6	

A bumping route has at most one box in each of several successive rows, starting at the top. We say that a route R is ***strictly left*** (resp. ***weakly left***) of a route R' if in each row which contains a box of R', R has a box which is left of (resp. left of or equal to) the box in R'. We use corresponding strict and weak terminology for positions above or below a given box or row.

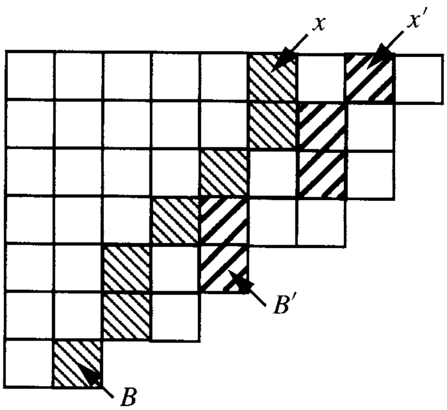
Row Bumping Lemma *Consider two successive row-insertions, first row-inserting x in a tableau T and then row-inserting x' in the resulting tableau $T \leftarrow x$, giving rise to two routes R and R', and two new boxes B and B'.*

(1) *If $x \le x'$, then R is strictly left of R', and B is strictly left of and weakly below B'.*

(2) *If $x > x'$, then R' is weakly left of R and B' is weakly left of and strictly below B.*



Proof This is a question of keeping track of what happens as the elements bump through a given row. Suppose $x \le x'$, and x bumps an element y from the first row. The element y' bumped by x' from the first row must lie strictly to the right of the box where x bumped, since the elements in that box or to the left are no larger than x. In particular, $y \le y'$, and the same argument continues from row to row. Note that the route for R cannot stop above that of R', and if R' stops first, the route for R never moves to the right, so the box B must be strictly left of and weakly below B'.



On the other hand, if $x > x'$, and x and x' bump elements y and y', respectively, the box in the first row where the bumping occurs for x' must be at or to the left of the box where x bumped, and in either case, we must have $y > y'$, so the argument can be repeated on successive rows. This time the route R' must continue at least one row below that of R. □

This lemma has the following important consequence.

Proposition *Let T be a tableau of shape λ, and let*

$$U = ((T \leftarrow x_1) \leftarrow x_2) \leftarrow \ldots \leftarrow x_p,$$

for some $x_1, \ldots, x_p$. Let μ be the shape of U. If $x_1 \leq x_2 \leq \ldots \leq x_p$ (resp. $x_1 > x_2 > \ldots > x_p$), then no two of the boxes in μ/λ are in the same column (resp. row). Conversely, suppose U is a tableau on a shape μ, and λ a Young diagram contained in μ, with p boxes in μ/λ. If no two boxes in μ/λ are in the same column (resp. row), then there is a unique tableau T of shape λ, and unique $x_1 \leq x_2 \leq \ldots \leq x_p$ (resp. $x_1 > x_2 > \ldots > x_p$) such that $U = ((T \leftarrow x_1) \leftarrow x_2) \leftarrow \ldots \leftarrow x_p$.

Proof The first assertion is a direct consequence of the lemma. For the converse, in the case where μ/λ has no two boxes in the same column, do reverse row bumping on U, using the boxes in μ/λ, starting from the right-most box and working to the left. The tableau T is the tableau obtained after these operations are carried out, and $x_p, \ldots, x_1$ are the elements bumped out. The Row Bumping Lemma guarantees that the resulting sequence satisfies $x_1 \leq \ldots \leq x_p$. Similarly, if μ/λ has no two boxes in the same row, do p reverse bumpings, starting from the lowest box in μ/λ, and working up; again, the Row Bumping Lemma implies that the elements $x_p, \ldots, x_1$ bumped out satisfy $x_1 > x_2 > \ldots > x_p$. □

This Schensted operation has many remarkable properties. It can be used to form a ***product tableau*** $T \cdot U$ from any two tableaux T and U. The number of boxes in the product will be the sum of the number of boxes in each, and its entries will be the entries of T and U. If U consists of one box with entry x, the product $T \cdot U$ is the result $T \leftarrow x$ of row-inserting x in T. To construct it in general, start with T, and row-insert the left-most entry in the bottom row of U into T. Row-insert into the result the next entry of the bottom row of U, and continue until all entries in the bottom row of U have been inserted. Then insert in order the entries of the next to the last row, left to right, and continue with the other rows, until all the entries of U have been inserted. In other words, if we list the entries of U in order from left to right, and from bottom to top, getting a sequence $x_1, x_2, \ldots, x_s$, then

$$T \cdot U = ((\ldots((T \leftarrow x_1) \leftarrow x_2) \leftarrow \ldots) \leftarrow x_{s-1}) \leftarrow x_s.$$